**VERİ MADENCİLİĞİNE GİRİŞ**

**RAPOR**

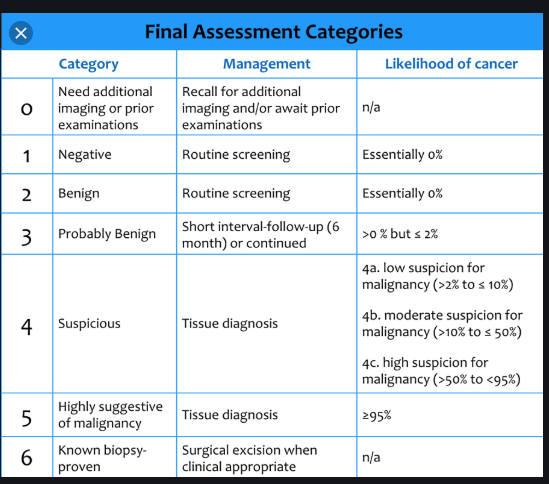
**Ömer Faruk Yavuz**

**17011083**

**Understanding the Data**

Bu raporda Mammographic Mass veri setinin analizinin anlatılması amaçlanmıştır. Veri setinde toplam 951 tane instance ve 6 tane attribute bulunmaktadır. Attribute’lar hakkında özellikler aşağıda detaylıca açıklanmıştır:

1. BI-RADS: Breast Imaging-Reporting and Data System’in kısaltmasıdır. Bu kategorilerde 0’dan 6’ya kadar olan sınıflandırmalar vardır. Aşağıdaki resim bu kısaltmaların ne için kullanıldığını göstermektedir.



BI-RADS’larda normalde 0’dan 6’ya kadar değer olması gerekirken, sonuçlarda n/a ifadeleri (not available) olanlar veri setine dahil edilmemiştir. 0-6 yerine 1-5 arasındaki değerler kullanılmıştır.

**Attribute’lar Hakkında Bilgiler:**

1. BI-RADS ölçeklendirmesi: 1 to 5 (ordinal)

2. Age: hastanın yaşı (integer)

3. Shape: mass shape: round=1 oval=2 lobular=3 irregular=4 (nominal)

4. Margin: mass margin: circumscribed=1 microlobulated=2 obscured=3 ill-defined=4 spiculated=5 (nominal)

5. Density: mass density high=1 iso=2 low=3 fat-containing=4 (ordinal)

6. Severity: benign=0 or malignant=1 (binominal) (iyi niyetli – kötü niyetli) (class olarak etiketlenecektir).

Verilerin sınıflandırılması sonucunda, iyi huylu hücrelere (benign) sahip hastaların sayısının 516, kötücül hücrelere (malignant) sahip olan hastaların sayısının 445 olduğunu görmekteyiz.

Projede birçok kayıp değer bulunmaktadır. Bu değerler veri setinde ‘?’ ile gösterilmiş olup sayıları aşağıda gösterildiği şekildedir:

- BI-RADS assessment: 2

- Age: 5

- Shape: 31

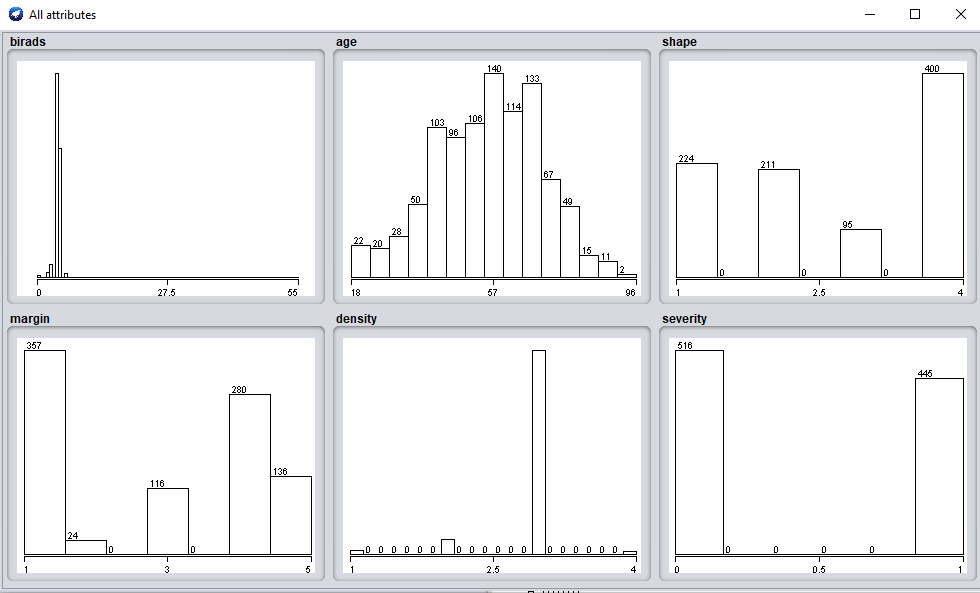
- Margin: 48

- Density: 76

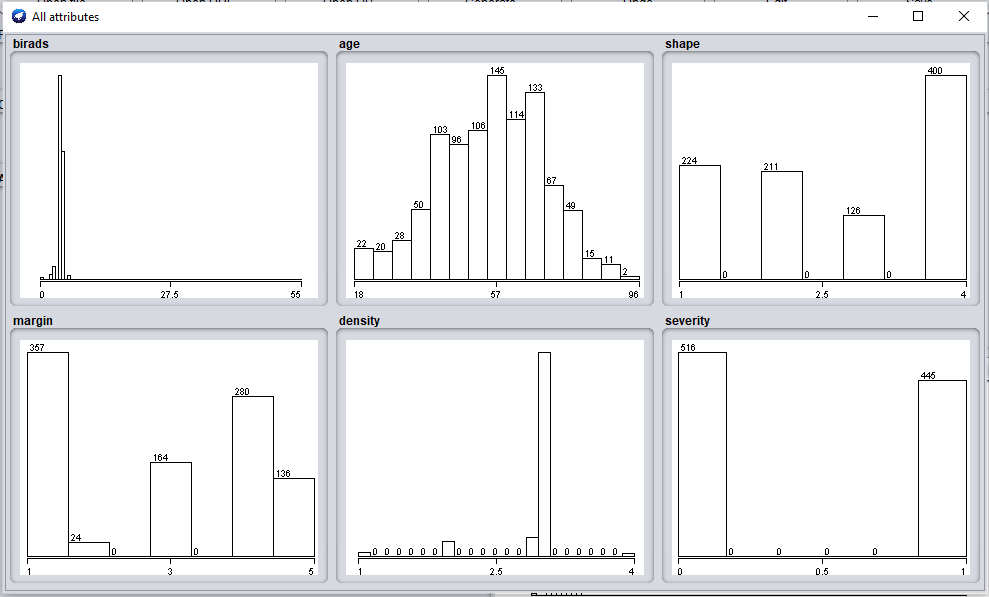
- Severity: 0

Verilerle çalışmadan önce, elimizdeki .data dosyası .csv dosyasına, daha sonra da bu dosya .arff dosyasına çevirilmiştir. Böylelikle WEKA ortamında çalışma imkanı elde edilmiştir.

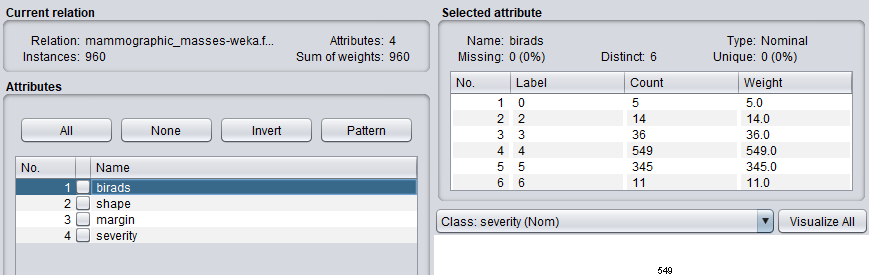
Kayıp değerlerin de dahil olduğu attribute’ların histogramları aşağıdaki ekran görüntüsünde verilmiştir:



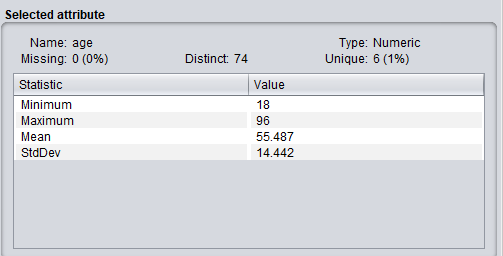
Kayıp değerleri ortadan kaldırmak için, Weka’da filtreleme işlemi uygulanmıştır. Choose butonundan “filters” 🡪 “unsupervised” 🡪 “attribute” 🡪 “replace missing values” seçenekleri uygulanmış olup kayıp değerler ortadan kaldırıldığında, attributeların sahip olduğu histogramlar aşağıdaki şekilde gösterilmektedir

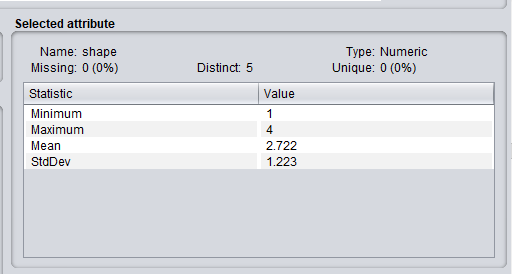


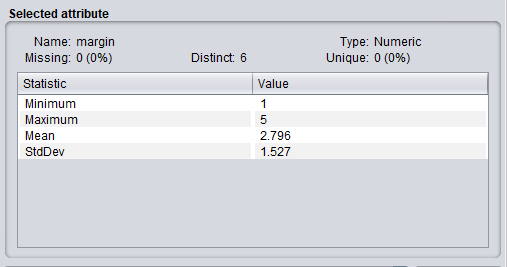
Her bir attribute’taki özellikler, sağ taraftaki “Selected Attribute” kısmında görülebilmektedir. Görüldüğü üzere “Missing: 0” belirteciyle bu veride hiçbir missing value’nun olmadığı görülmektedir.

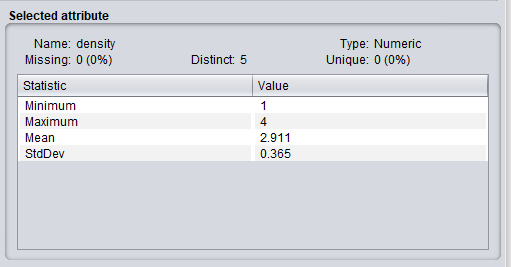


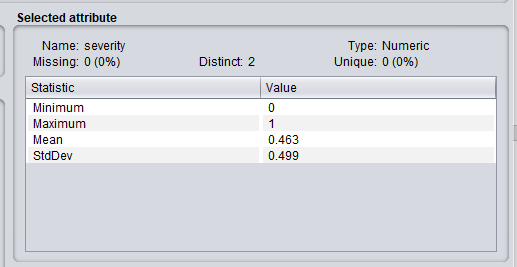
Her bir attribute için yeni istatistiksel bilgiler aşağıdaki ekran görüntülerinde belirtilmiştir. Seçilen attribute’lerin isimleri ve tipleri aşağıda gösterildiği şekildedir. Kayıp değer (missing value), uygulanan filtre işleminden sonra sıfıra indirgenmiştir. Attribute’ların sahip oldukları distinct ve unique değerler aşağıda gösterilen şekildedir







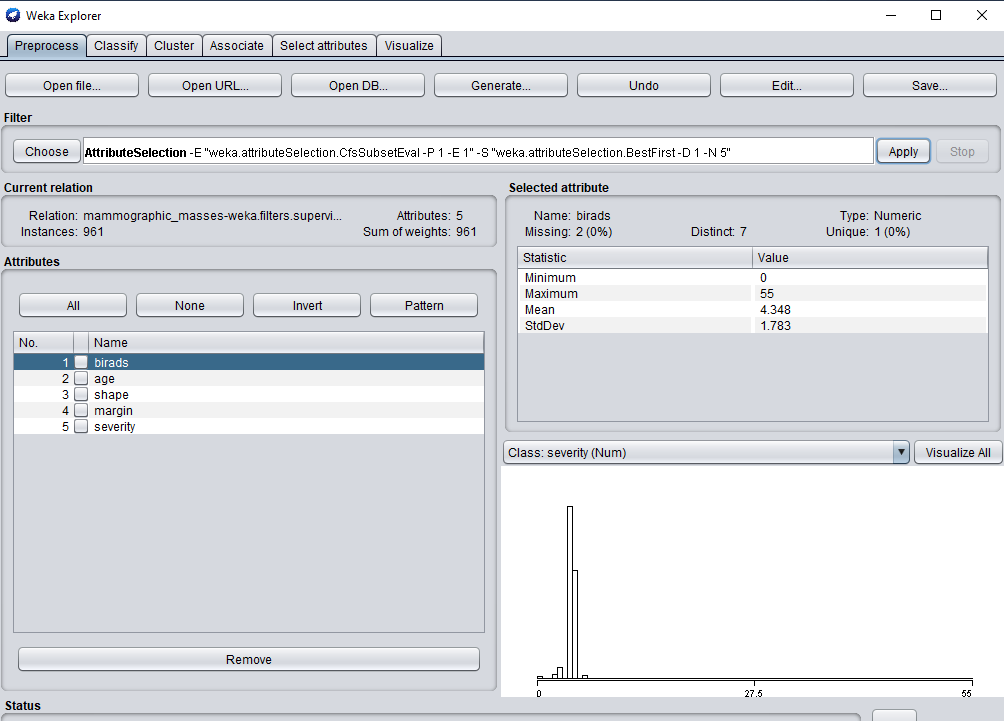




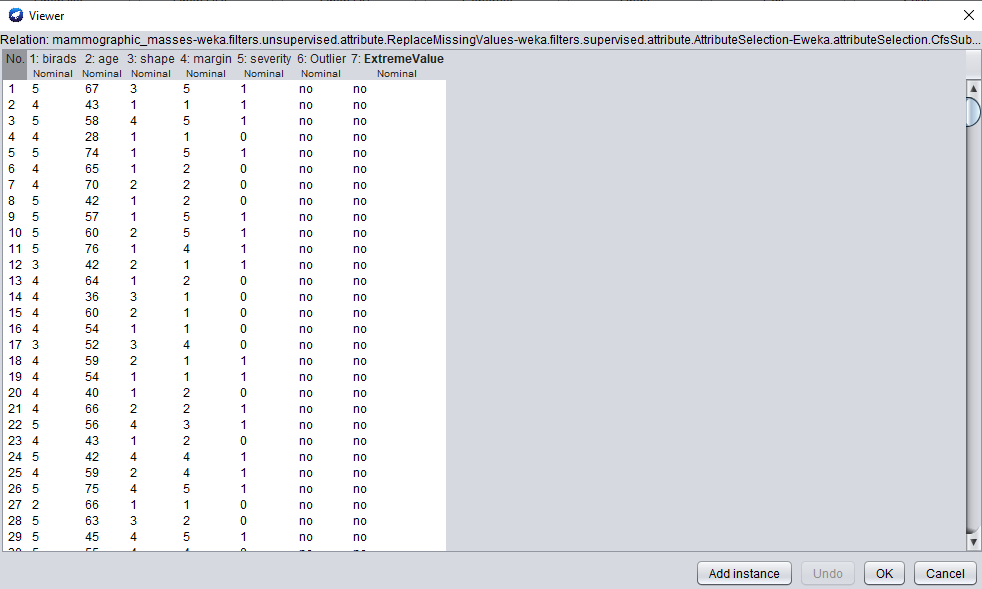
Veri setinde “age” ve “density” attribute’ları çıkartılmış olup veri madenciliği işlemlerinin daha da efektif hale gelmesi amaçlanmıştır.

**Applying Filters**

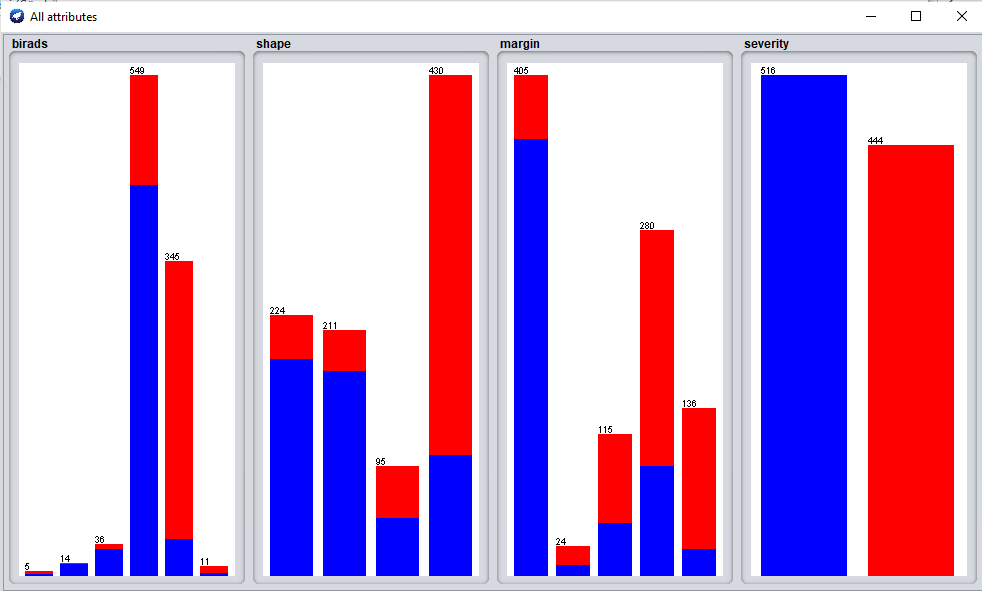
Sınıf etiketi için en iyi attribute’ların seçilmesini isteriz. **weka→filters→supervised→attribute→AttributeSelection** işlemleri ile bizim için en iyi kullanılabilecek attribute’lar belirlenmiş oldu. Aşağıdaki ekran görüntüsünde seçilen önemli attribute’ların listesi görülmektedir.

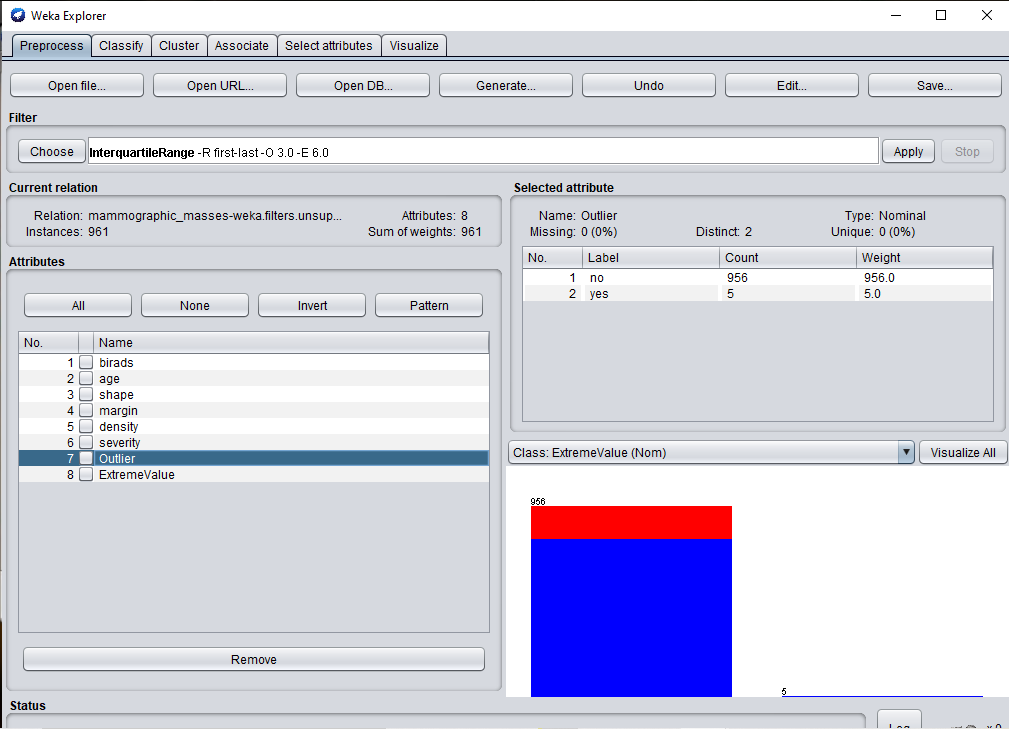


Doldurulan eksik değerlerin ekran görüntüsü aşağıdaki şekilde verilmiştir

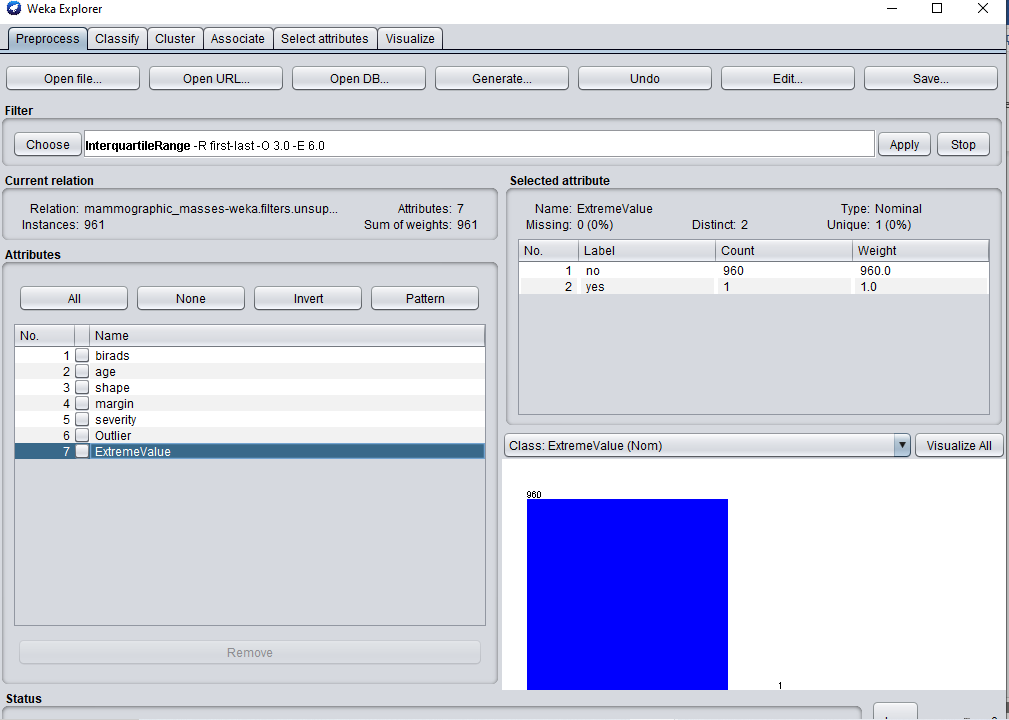


Histogramlar aşağıdaki ekran görüntüsünde gösterilmiştir

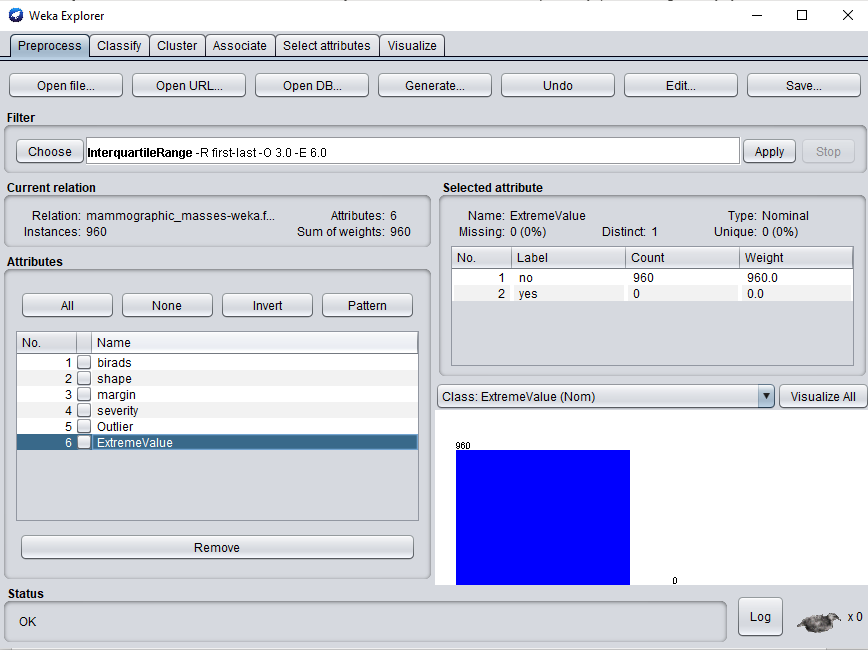


Aşağıdaki ekran görüntüsünde de görüldüğü üzere 5 değer outlier olarak sınıflandırılmıştır.

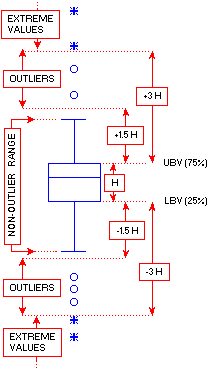
Bir değerin extreme value olduğu görülmektedir



İki attrbiute değeri çıkarttığımızda elde ettiğimiz Extreme Value’ya ait ekran görüntüsü aşağıdaki şekildedir

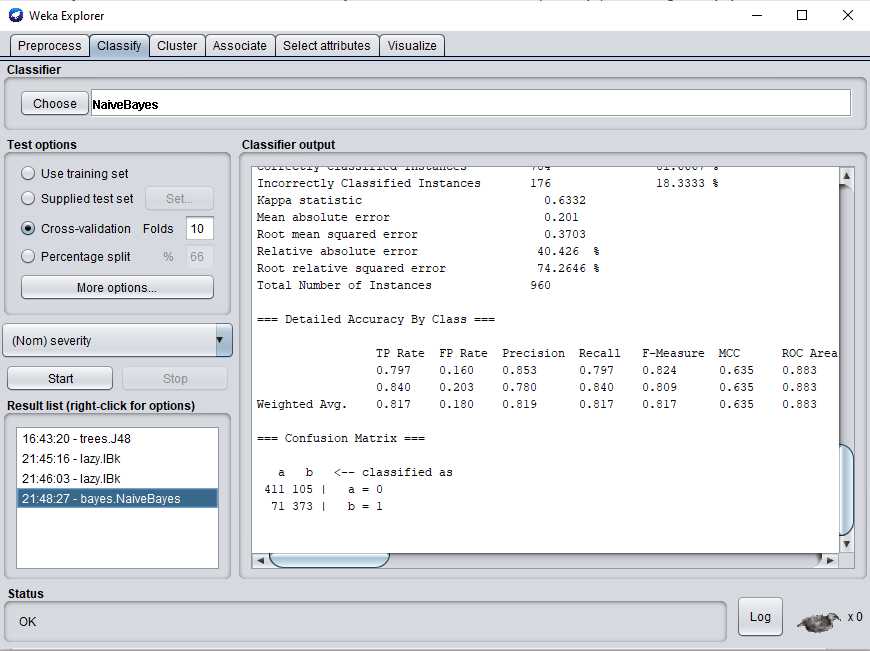


Extreme value ile outlier arasındaki fark aşağıdaki resimle rahatça anlaşılabilir



**Naïve Bayes Classifier**

Elimizdeki numeric değerleri WEKA’da nominal değerlere çevirdik. Unsupervised attribute’tan Numeric to Nominal yaptığımızda bu değerleri nominal hale getirilmiştir. Daha sonra bu nominal değerler Bayes sınıflandırmada kullanılmıştır. Elde edilen sonuçlar aşağıdaki ekran görüntüsünde gösterilmiştir



Doğru bir şekilde sınıflandırılan instance’ların oranı %81 iken yanlış sınıflandırılanların oranı %18 olmuştur. Class confusion matrix yine ekran görüntüsünde verilmiştir.

Elde edilen ekran çıktısı aşağıda gösterilmektedir

=== Run information ===

Scheme: weka.classifiers.bayes.NaiveBayes

Relation: mammographic\_masses-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.supervised.attribute.AttributeSelection-Eweka.attributeSelection.CfsSubsetEval -P 1 -E 1-Sweka.attributeSelection.BestFirst -D 1 -N 5-weka.filters.unsupervised.attribute.ReplaceMissingValues-weka.filters.unsupervised.attribute.InterquartileRange-Rfirst-last-O3.0-E6.0

Instances: 960

Attributes: 6

birads

shape

margin

severity

Outlier

ExtremeValue

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class

Attribute 0 1

(0.54) (0.46)

=============================

birads

0 3.0 4.0

2 14.0 2.0

3 31.0 7.0

4 429.0 122.0

5 41.0 306.0

6 4.0 9.0

[total] 522.0 450.0

shape

1 187.0 39.0

2 177.0 36.0

3 51.0 46.0

4 105.0 327.0

[total] 520.0 448.0

margin

1 354.0 53.0

2 10.0 16.0

3 44.0 73.0

4 90.0 192.0

5 23.0 115.0

[total] 521.0 449.0

Outlier

no 517.0 445.0

yes 1.0 1.0

[total] 518.0 446.0

ExtremeValue

no 517.0 445.0

yes 1.0 1.0

[total] 518.0 446.0

Time taken to build model: 0.04 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 784 81.6667 %

Incorrectly Classified Instances 176 18.3333 %

Kappa statistic 0.6332

Mean absolute error 0.201

Root mean squared error 0.3703

Relative absolute error 40.426 %

Root relative squared error 74.2646 %

Total Number of Instances 960

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.797 0.160 0.853 0.797 0.824 0.635 0.883 0.861 0

0.840 0.203 0.780 0.840 0.809 0.635 0.883 0.836 1

Weighted Avg. 0.817 0.180 0.819 0.817 0.817 0.635 0.883 0.850

=== Confusion Matrix ===

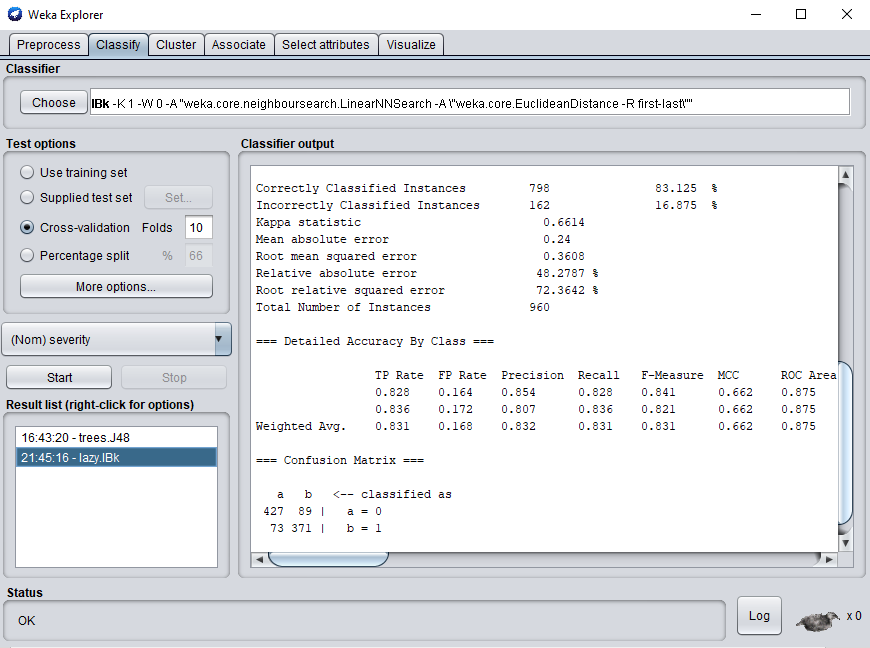
a b <-- classified as

411 105 | a = 0

71 373 | b = 1

**K-Nearest Neighbour Classifier**

Bu sınıflandırma aracı ile K-Nearest Neighbour sınıflayıcı kullanılmıştır. Yukarıdaki ekran görüntüsünde belirtildiği üzere başarı oranı %83’dir. Confusion Matrix de yine ekran görüntüsünde belirtilmiştir. Bu sınıflandırmada KNN = 1 alınmıştır.



Örneğin KNN = 9 için sınıflandırma yaptığımızda başarı oranının %81’e düştüğü görülmüştür.

KNN = 1 için sonuç çıktısı aşağıda verildiği şekildedir

=== Run information ===

Scheme: weka.classifiers.lazy.IBk -K 9 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Relation: mammographic\_masses-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.supervised.attribute.AttributeSelection-Eweka.attributeSelection.CfsSubsetEval -P 1 -E 1-Sweka.attributeSelection.BestFirst -D 1 -N 5-weka.filters.unsupervised.attribute.ReplaceMissingValues-weka.filters.unsupervised.attribute.InterquartileRange-Rfirst-last-O3.0-E6.0

Instances: 960

Attributes: 6

birads

shape

margin

severity

Outlier

ExtremeValue

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

IB1 instance-based classifier

using 9 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 784 81.6667 %

Incorrectly Classified Instances 176 18.3333 %

Kappa statistic 0.6322

Mean absolute error 0.255

Root mean squared error 0.367

Relative absolute error 51.2819 %

Root relative squared error 73.6044 %

Total Number of Instances 960

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

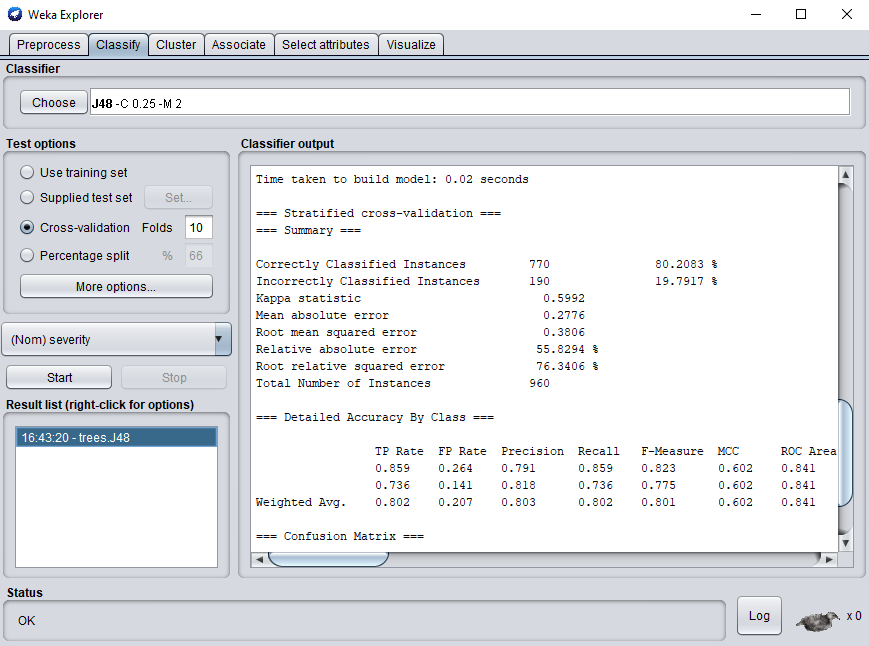
0.814 0.180 0.840 0.814 0.827 0.633 0.879 0.877 0

0.820 0.186 0.791 0.820 0.805 0.633 0.879 0.842 1

Weighted Avg. 0.817 0.183 0.817 0.817 0.817 0.633 0.879 0.861

=== Confusion Matrix ===

**Decision Tree Classifier**



Elde edilen sonuç aşağıdaki şekildedir

=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: mammographic\_masses-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.supervised.attribute.AttributeSelection-Eweka.attributeSelection.CfsSubsetEval -P 1 -E 1-Sweka.attributeSelection.BestFirst -D 1 -N 5-weka.filters.unsupervised.attribute.ReplaceMissingValues-weka.filters.unsupervised.attribute.InterquartileRange-Rfirst-last-O3.0-E6.0

Instances: 960

Attributes: 6

birads

shape

margin

severity

Outlier

ExtremeValue

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

------------------

birads = 0: 1 (5.0/2.0)

birads = 2: 0 (14.0/1.0)

birads = 3: 0 (36.0/6.0)

birads = 4

| shape = 1: 0 (184.0/17.0)

| shape = 2: 0 (173.0/17.0)

| shape = 3: 0 (55.0/15.0)

| shape = 4

| | margin = 1: 0 (13.0/5.0)

| | margin = 2: 0 (4.0/2.0)

| | margin = 3: 0 (17.0/6.0)

| | margin = 4: 1 (75.0/34.0)

| | margin = 5: 1 (28.0/10.0)

birads = 5: 1 (345.0/40.0)

birads = 6

| margin = 1: 0 (3.0)

| margin = 2: 1 (0.0)

| margin = 3: 1 (4.0)

| margin = 4: 1 (2.0)

| margin = 5: 1 (2.0)

Number of Leaves : 17

Size of the tree : 21

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 770 80.2083 %

Incorrectly Classified Instances 190 19.7917 %

Kappa statistic 0.5992

Mean absolute error 0.2776

Root mean squared error 0.3806

Relative absolute error 55.8294 %

Root relative squared error 76.3406 %

Total Number of Instances 960

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.859 0.264 0.791 0.859 0.823 0.602 0.841 0.826 0

0.736 0.141 0.818 0.736 0.775 0.602 0.841 0.797 1

Weighted Avg. 0.802 0.207 0.803 0.802 0.801 0.602 0.841 0.812

=== Confusion Matrix ===

a b <-- classified as

443 73 | a = 0

117 327 | b = 1

Ağacın boyutu: 21

TP değeri: 443

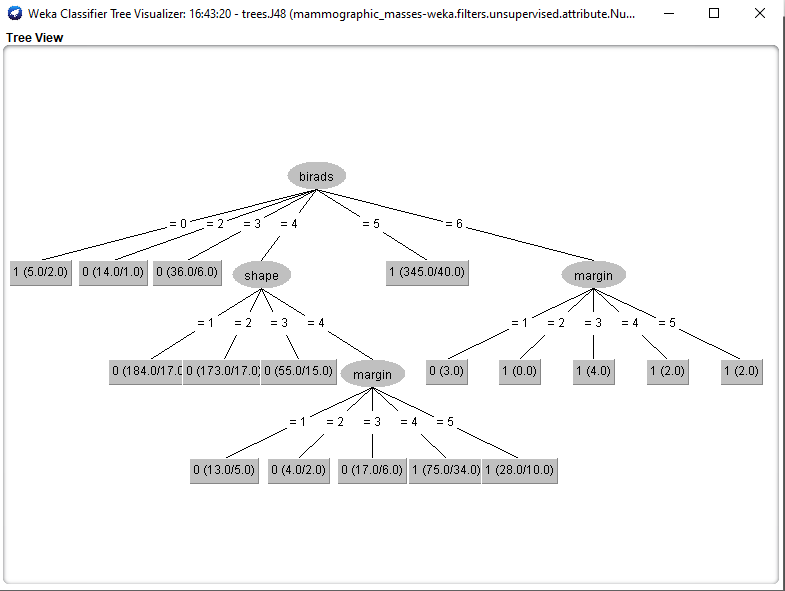
TN değeri: 327

FT değeri: 73

FP değeri: 117

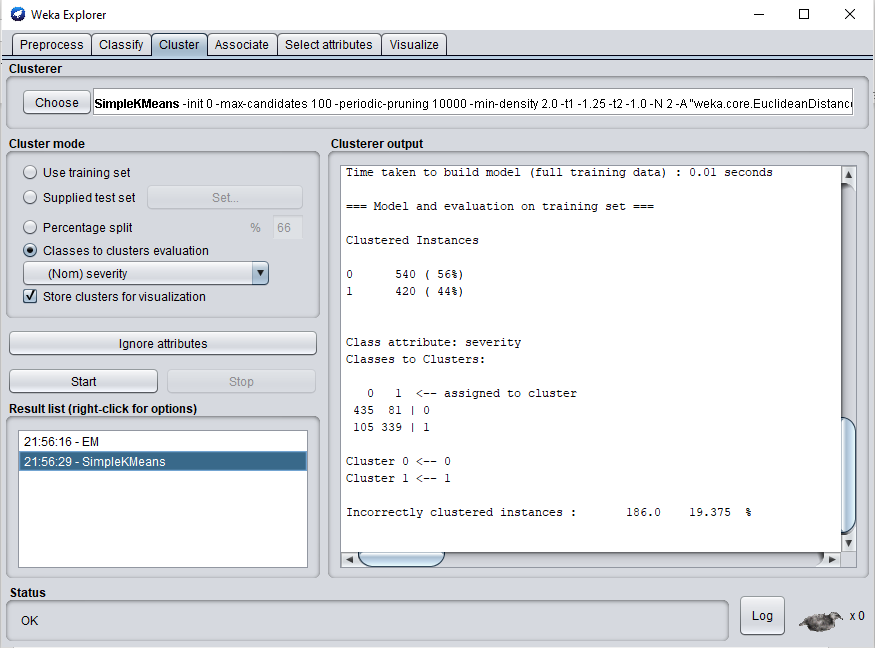
Yukarıdaki örnekte Karar Ağaçları metodu kullanılmıştır. Burada %80 oranında başarı oranı elde edilmiştir. Class Confusion Matrix ekran görüntüsünde belirtildiği şekildedir. Elde edilen üç sınıflandırmaya göre en başarılı sınıflandırma Karar Ağaçları metodu olmuştur.

Aşağıdaki görselde karar ağacı gösterilmektedir



**K-Means Clustering**

Projede K-Means kümeleme metodu kullanılmıştır. Elde edilen ekran görüntüsü aşağıdaki şekildedir



Elde edilen sonuç aşağıdaki gibidir

=== Run information ===

Scheme: weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Relation: mammographic\_masses-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.supervised.attribute.AttributeSelection-Eweka.attributeSelection.CfsSubsetEval -P 1 -E 1-Sweka.attributeSelection.BestFirst -D 1 -N 5-weka.filters.unsupervised.attribute.ReplaceMissingValues-weka.filters.unsupervised.attribute.InterquartileRange-Rfirst-last-O3.0-E6.0

Instances: 960

Attributes: 6

birads

shape

margin

Ignored:

severity

Outlier

ExtremeValue

Test mode: Classes to clusters evaluation on training data

=== Clustering model (full training set) ===

kMeans

======

Number of iterations: 4

Within cluster sum of squared errors: 865.0

Initial starting points (random):

Cluster 0: 4,3,3

Cluster 1: 5,2,4

Missing values globally replaced with mean/mode

Final cluster centroids:

Cluster#

Attribute Full Data 0 1

(960.0) (540.0) (420.0)

============================================

birads 4 4 5

shape 4 1 4

margin 1 1 4

Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0 540 ( 56%)

1 420 ( 44%)

Class attribute: severity

Classes to Clusters:

0 1 <-- assigned to cluster

435 81 | 0

105 339 | 1

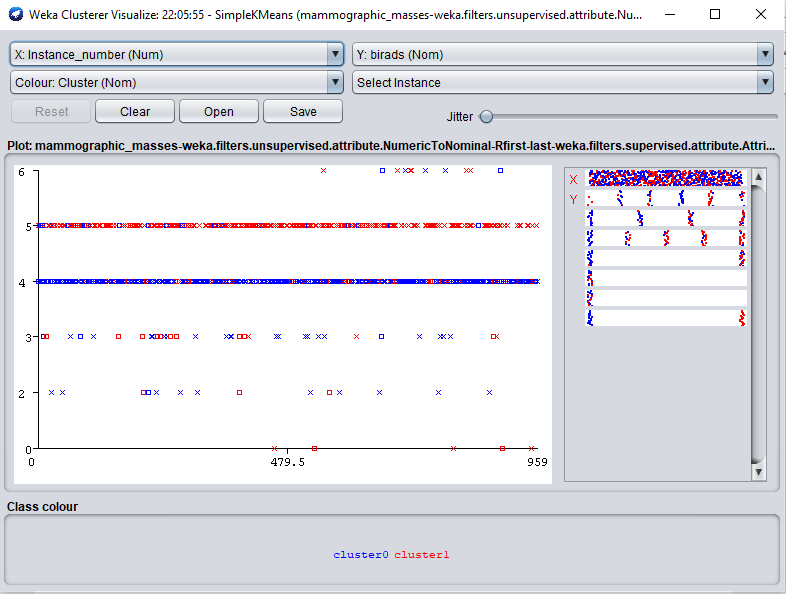
Cluster 0 <-- 0

Cluster 1 <-- 1

Incorrectly clustered instances : 186.0 19.375 %

İki farklı {0 ve 1} clustered instance’a sahibiz. 540 instance Cluster 0’a, 420 instance Cluster 1’e aittir. TP değeri = 435, TN değeri = 339, FP değeri = 81 FN değeri = 105’tir.

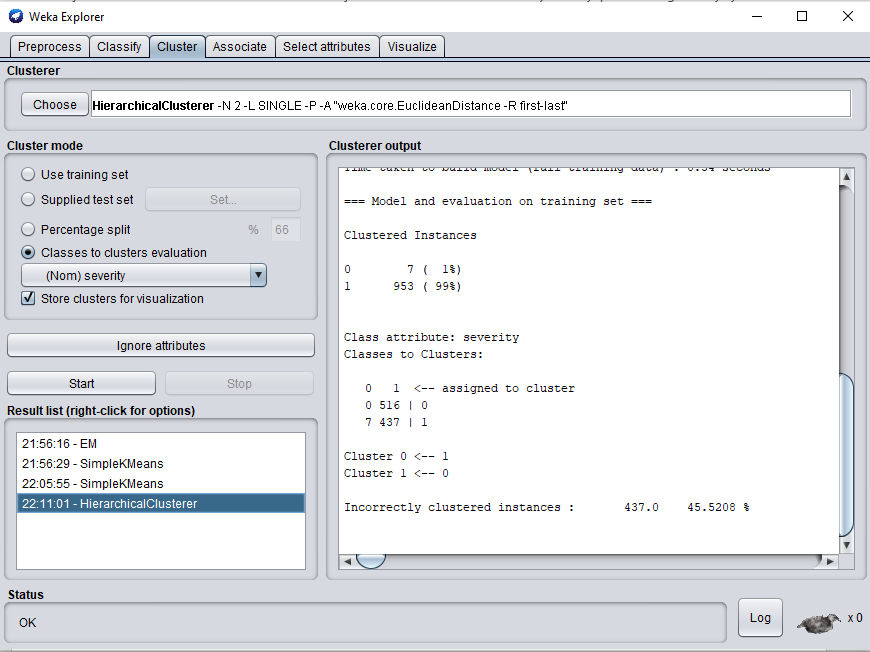
Elde ettiğimiz kümeleme verilerini görselleştirdiğimizde elde ettiğimiz sonuç aşağıdaki ekran görüntüsünde verilmiştir



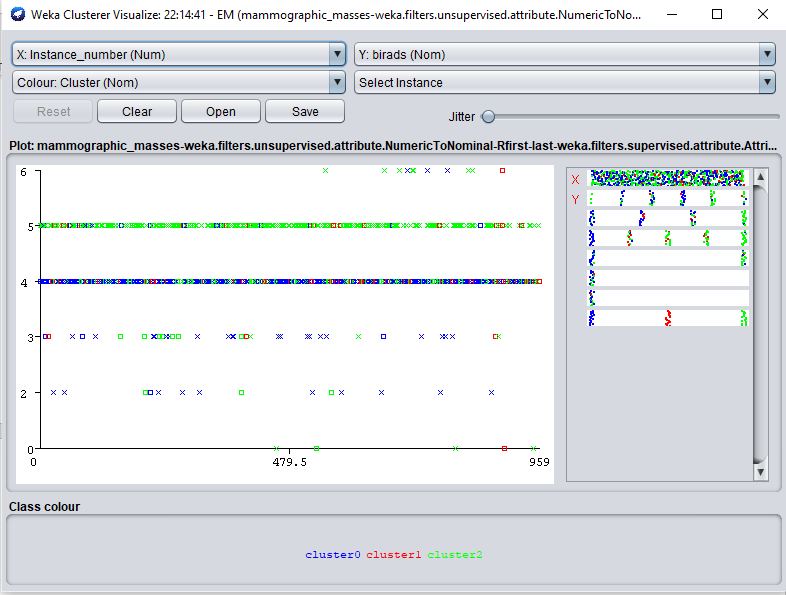
Elimizde 284, 395, 20, 63, 199’er adet instance’lara sahip olan 5 adet cluster vardır. Yukarıdaki görsele görüleceği üzere cluster0, cluster1 ve cluster4 iyi bir şekilde kümelenmiştir. Cluster2 ve cluster3’ün kümelemeleri ise daha ayrıktır. Hangi instance’ların hangi cluster’a ait olduğunu göstermek için bir filtreleme işlemi uygulanmaktadır. Kullanılan filtre işlemi bir unsupervised filter’dır. Clustering işlemi 6 attribute’a bakılarak yapılmaktadır. K=2 için 2 farklı küme vardır. Örneklerin ait olduğu kümelerin listesinin bir örneği aşağıdaki ekran görüntüsünde verilmiştir

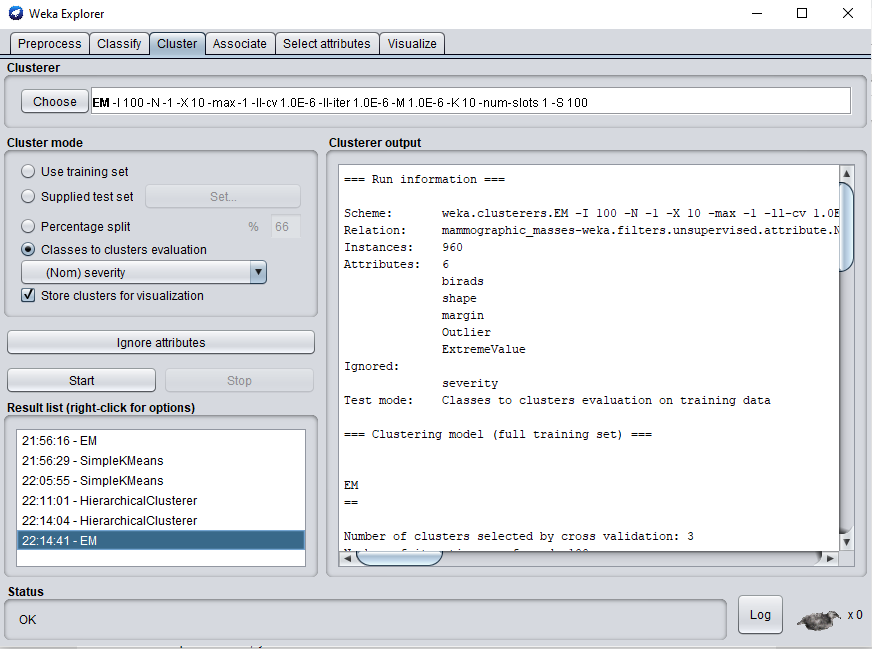
**Hierarchical Clusterer**

Ekran görüntüsü



**EM Clustering**





=== Run information ===

Scheme: weka.clusterers.EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100

Relation: mammographic\_masses-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.supervised.attribute.AttributeSelection-Eweka.attributeSelection.CfsSubsetEval -P 1 -E 1-Sweka.attributeSelection.BestFirst -D 1 -N 5-weka.filters.unsupervised.attribute.ReplaceMissingValues-weka.filters.unsupervised.attribute.InterquartileRange-Rfirst-last-O3.0-E6.0

Instances: 960

Attributes: 6

birads

shape

margin

Outlier

ExtremeValue

Ignored:

severity

Test mode: Classes to clusters evaluation on training data

=== Clustering model (full training set) ===

EM

==

Number of clusters selected by cross validation: 3

Number of iterations performed: 100

Cluster

Attribute 0 1 2

(0.42) (0.12) (0.47)

==========================================

birads

0 1.0108 2.1627 4.8265

2 11.6743 1.404 3.9217

3 23.2121 4.6446 11.1432

4 347.8731 79.227 124.8998

5 18.3933 29.6365 299.9702

6 3.4478 2.759 7.7932

[total] 405.6115 119.8339 452.5546

shape

1 207.3935 2.9654 16.6411

2 154.9643 50.2494 8.7863

3 25.3906 44.0431 28.5663

4 15.863 20.5761 396.5609

[total] 403.6115 117.8339 450.5546

margin

1 384.1254 18.9103 4.9643

2 5.1172 9.0126 12.8702

3 8.1056 28.8023 81.0921

4 4.7735 60.0931 218.1333

5 2.4898 2.0156 134.4946

[total] 404.6115 118.8339 451.5546

Outlier

no 400.6115 114.8339 447.5546

yes 1 1 1

[total] 401.6115 115.8339 448.5546

ExtremeValue

no 400.6115 114.8339 447.5546

yes 1 1 1

[total] 401.6115 115.8339 448.5546

Time taken to build model (full training data) : 3.19 seconds

=== Model and evaluation on training set ===

Clustered Instances

0 405 ( 42%)

1 97 ( 10%)

2 458 ( 48%)

Log likelihood: -3.02507

Class attribute: severity

Classes to Clusters:

0 1 2 <-- assigned to cluster

354 63 99 | 0

51 34 359 | 1

Cluster 0 <-- 0

Cluster 1 <-- No class

Cluster 2 <-- 1

Incorrectly clustered instances : 247.0 25.7292 %

MATLAB Code for Decision Tree

%Author: Ömer Faruk Yavuz

%Project for Data Mining Lecture.

%Instructor: Songül Varl?.

clc;

clear all;

close all;

import java.util.ArrayList

%Importing the data

data = readmatrix('mammographic\_masses\_new4.csv');

%Split the data as train and test

%K-Fold Cross Validation

%Data has been split up into 10 K-Fold Cross Validation, Severity is the

%class attribute.

cv = cvpartition(data(:, 6),'KFold',10);

idxTest = test(cv, 1); %Test indices for cross-validation

idxTraining = training(cv, 10); %Training indices for cross-validation

%Separate to training and test data

dataTrain = data(~idxTraining,:);

dataTest = data(idxTest,:);

%Max information gain

maxInfo=0;

%Splitting the data into attributes

BIRADS = data(:, 1);

Age = data(:, 2);

Shape = data(:, 3);

Margin = data(:, 4);

Density = data(:, 5);

Severity = data(:, 6);

attributeArray=ArrayList;

attributeArray.add('BIRADS');

attributeArray.add('Age');

attributeArray.add('Shape');

attributeArray.add('Margin');

attributeArray.add('Density');

attributeArray.add('Severity');

BIRADSValues=ArrayList;

for i=1:960

if(BIRADSValues.contains(BIRADS(i)) == 0)

BIRADSValues.add(BIRADS(i));

end

end

BIRADSValues;

AgeValues = ArrayList;

for i=1:960

if(AgeValues.contains(Age(i)) == 0)

AgeValues.add(Age(i));

end

end

AgeValues;

ShapeValues = ArrayList;

for i=1:960

if(ShapeValues.contains(Shape(i)) == 0)

ShapeValues.add(Shape(i));

end

end

ShapeValues;

MarginValues = ArrayList;

for i=1:960

if(MarginValues.contains(Margin(i)) == 0)

MarginValues.add(Margin(i));

end

end

MarginValues;

DensityValues = ArrayList;

for i=1:960

if(DensityValues.contains(Density(i)) == 0)

DensityValues.add(Density(i));

end

end

DensityValues;

%Class Label

SeverityValues = ArrayList;

for i=1:960

if(SeverityValues.contains(Severity(i)) == 0)

SeverityValues.add(Severity(i));

end

end

SeverityValues;

%Gain Calculation

yes=0;

no=0;

for i=1:960

if(Severity(i) == 1)

yes=yes+1;

else

no=no+1;

end

end

disp("Sinif etiketi için Evet sayisi: " + yes);

disp("Sinif etiketi için Hayir sayisi: " + no);

info = -(yes/(yes+no)\*log2(yes/(yes+no))) -(no/(yes+no)\*log2(no/(yes+no)));

disp("Info(D)=" + info);

disp("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

disp("BIRADS için evet/hayir oranlari");

countYes=0;

countNo=0;

entropy=0;

totalEntropy=0;

for i=1:BIRADSValues.size()

tempVal = BIRADSValues.get(i-1);

countYes=0;

countNo=0;

for k=1:960

if(Severity(k)==1 && tempVal==BIRADS(k))

countYes=countYes+1;

elseif(Severity(k)==0 && tempVal==BIRADS(k))

countNo=countNo+1;

end

end

entropy = ((countYes+countNo) / 960)\*(-countYes/(countYes+countNo)\*log2(countYes/(countYes+countNo)) -countNo/(countYes+countNo)\*log2(countNo/(countYes+countNo)));

disp("BIRADS="+tempVal+" degeri için");

disp("Yes: " + countYes);

disp("No: " + countNo);

disp("Info\_birads(D): " + entropy);

totalEntropy = totalEntropy + entropy;

end

disp("Information Gain=" + (info-totalEntropy));

if(maxInfo<(info-totalEntropy))

maxInfo = (info-totalEntropy);

end

disp("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

disp("Age için evet/hayir oranlari");

countYes=0;

countNo=0;

entropy=0;

totalEntropy=0;

for i=1:AgeValues.size()

tempVal = AgeValues.get(i-1);

countYes=0;

countNo=0;

for k=1:960

if(Severity(k)==1 && tempVal==Age(k))

countYes=countYes+1;

elseif(Severity(k)==0 && tempVal==Age(k))

countNo=countNo+1;

end

end

if(countYes ~= 0 && countNo ~= 0)

entropy = ((countYes+countNo) / 960)\*(-countYes/(countYes+countNo)\*log2(countYes/(countYes+countNo)) -countNo/(countYes+countNo)\*log2(countNo/(countYes+countNo)));

disp("Yes: " + countYes);

disp("No: " + countNo);

disp("Info\_age(D): " + entropy);

totalEntropy = totalEntropy + entropy;

end

end

disp("Information Gain=" + (info-totalEntropy));

if(maxInfo<(info-totalEntropy))

maxInfo = (info-totalEntropy);

end

disp("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

disp("Density için evet/hayir oranlari");

countYes=0;

countNo=0;

entropy=0;

totalEntropy=0;

for i=1:DensityValues.size()

tempVal = DensityValues.get(i-1);

countYes=0;

countNo=0;

for k=1:960

if(Severity(k)==1 && tempVal==Density(k))

countYes=countYes+1;

elseif(Severity(k)==0 && tempVal==Density(k))

countNo=countNo+1;

end

end

if(countYes ~= 0 && countNo ~= 0)

entropy = ((countYes+countNo) / 960)\*(-countYes/(countYes+countNo)\*log2(countYes/(countYes+countNo)) -countNo/(countYes+countNo)\*log2(countNo/(countYes+countNo)));

disp("Yes: " + countYes);

disp("No: " + countNo);

disp("Info\_density(D): " + entropy);

totalEntropy = totalEntropy + entropy;

end

end

disp("Information Gain=" + (info-totalEntropy));

if(maxInfo<(info-totalEntropy))

maxInfo = (info-totalEntropy);

end

disp("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

disp("Margin için evet/hayir oranlari");

countYes=0;

countNo=0;

entropy=0;

totalEntropy=0;

for i=1:MarginValues.size()

tempVal = MarginValues.get(i-1);

countYes=0;

countNo=0;

for k=1:960

if(Severity(k)==1 && tempVal==Margin(k))

countYes=countYes+1;

elseif(Severity(k)==0 && tempVal==Margin(k))

countNo=countNo+1;

end

end

if(countYes ~= 0 && countNo ~= 0)

entropy = ((countYes+countNo) / 960)\*(-countYes/(countYes+countNo)\*log2(countYes/(countYes+countNo)) -countNo/(countYes+countNo)\*log2(countNo/(countYes+countNo)));

disp("Yes: " + countYes);

disp("No: " + countNo);

disp("Info\_margin(D): " + entropy);

totalEntropy = totalEntropy + entropy;

end

end

disp("Information Gain=" + (info-totalEntropy));

if(maxInfo<(info-totalEntropy))

maxInfo = (info-totalEntropy);

end

disp("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

disp("Shape için evet/hayir oranlari");

countYes=0;

countNo=0;

entropy=0;

totalEntropy=0;

for i=1:ShapeValues.size()

tempVal = ShapeValues.get(i-1);

countYes=0;

countNo=0;

for k=1:960

if(Severity(k)==1 && tempVal==Shape(k))

countYes=countYes+1;

elseif(Severity(k)==0 && tempVal==Shape(k))

countNo=countNo+1;

end

end

if(countYes ~= 0 && countNo ~= 0)

entropy = ((countYes+countNo) / 960)\*(-countYes/(countYes+countNo)\*log2(countYes/(countYes+countNo)) -countNo/(countYes+countNo)\*log2(countNo/(countYes+countNo)));

disp("Yes: " + countYes);

disp("No: " + countNo);

disp("Info\_shape(D): " + entropy);

totalEntropy = totalEntropy + entropy;

end

end

disp("Information Gain=" + (info-totalEntropy));

if(maxInfo<(info-totalEntropy))

maxInfo = (info-totalEntropy);

end

disp("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

disp("Maximum information gain=" + maxInfo);

disp("Attribute: BIRADS");

%Root is in the BIRADS.

disp("Root: BIRADS");

count=0;

tempArray = [];

severityTempArray = [];

indis=1;

for j=1:960

%By changing the value "6",...

%you can decide the leaves of the decision tree .

if(BIRADS(j) == 6)

disp(data(j, :));

tempArray(indis, :) = data(j,:);

severityTempArray(indis) = Severity(j);

count=count+1;

indis=indis+1;

end

end

count;

index=1;

for j=3:5

isOk=1;

for i=1:count-1

for k=i+1:count

%disp("tempArray("+i+","+j+")=" + tempArray(i,j) + " " + "tempArray("+k+","+j+")=" + tempArray(k,j))

%disp("severityTempArray("+i+")=" + severityTempArray(i) + " " + "severityTempArray("+k+")=" + severityTempArray(k))

if(tempArray(i,j) == tempArray(k,j) && severityTempArray(k) ~= severityTempArray(i))

isOk=0;

end

end

end

disp("ok: "+ isOk);

if(isOk == 1)

disp("Attribute: " + attributeArray.get(j-1));

end

end

count=0;

indis=1;

for i=1:960

if(BIRADS(i)==6)

disp(data(i,:))

count=count+1;

end

end

count

Çıktı:

Sinif etiketi için Evet sayisi: 444

Sinif etiketi için Hayir sayisi: 516

Info(D)=0.99594

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

BIRADS için evet/hayir oranlari

BIRADS=5 degeri için

Yes: 305

No: 40

Info\_birads(D): 0.18601

BIRADS=4 degeri için

Yes: 121

No: 428

Info\_birads(D): 0.43514

BIRADS=3 degeri için

Yes: 6

No: 30

Info\_birads(D): 0.024376

BIRADS=2 degeri için

Yes: 1

No: 13

Info\_birads(D): 0.0054138

BIRADS=0 degeri için

Yes: 3

No: 2

Info\_birads(D): 0.005057

BIRADS=6 degeri için

Yes: 8

No: 3

Info\_birads(D): 0.0096863

Information Gain=0.33026

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Age için evet/hayir oranlari

Yes: 23

No: 9

Info\_age(D): 0.028572

Yes: 7

No: 13

Info\_age(D): 0.01946

Yes: 14

No: 9

Info\_age(D): 0.023135

Yes: 1

No: 4

Info\_age(D): 0.00376

Yes: 11

No: 1

Info\_age(D): 0.0051727

Yes: 16

No: 9

Info\_age(D): 0.024549

Yes: 10

No: 3

Info\_age(D): 0.010554

Yes: 6

No: 13

Info\_age(D): 0.017807

Yes: 12

No: 20

Info\_age(D): 0.031814

Yes: 11

No: 14

Info\_age(D): 0.025771

Yes: 11

No: 6

Info\_age(D): 0.016587

Yes: 12

No: 15

Info\_age(D): 0.027874

Yes: 2

No: 9

Info\_age(D): 0.0078379

Yes: 12

No: 13

Info\_age(D): 0.026012

Yes: 11

No: 9

Info\_age(D): 0.020683

Yes: 22

No: 19

Info\_age(D): 0.042543

Yes: 7

No: 12

Info\_age(D): 0.018791

Yes: 19

No: 12

Info\_age(D): 0.031094

Yes: 8

No: 15

Info\_age(D): 0.022332

Yes: 11

No: 13

Info\_age(D): 0.024875

Yes: 4

No: 17

Info\_age(D): 0.015366

Yes: 14

No: 12

Info\_age(D): 0.026968

Yes: 7

No: 20

Info\_age(D): 0.023221

Yes: 2

No: 9

Info\_age(D): 0.0078379

Yes: 1

No: 1

Info\_age(D): 0.0020833

Yes: 6

No: 3

Info\_age(D): 0.008609

Yes: 5

No: 6

Info\_age(D): 0.01139

Yes: 3

No: 1

Info\_age(D): 0.0033803

Yes: 9

No: 12

Info\_age(D): 0.021552

Yes: 8

No: 5

Info\_age(D): 0.013017

Yes: 14

No: 11

Info\_age(D): 0.025771

Yes: 7

No: 11

Info\_age(D): 0.018076

Yes: 5

No: 19

Info\_age(D): 0.018457

Yes: 7

No: 16

Info\_age(D): 0.02124

Yes: 5

No: 11

Info\_age(D): 0.014934

Yes: 11

No: 6

Info\_age(D): 0.016587

Yes: 5

No: 4

Info\_age(D): 0.0092913

Yes: 15

No: 5

Info\_age(D): 0.016902

Yes: 2

No: 7

Info\_age(D): 0.0071644

Yes: 2

No: 11

Info\_age(D): 0.0083875

Yes: 2

No: 7

Info\_age(D): 0.0071644

Yes: 19

No: 3

Info\_age(D): 0.013169

Yes: 2

No: 1

Info\_age(D): 0.0028697

Yes: 4

No: 1

Info\_age(D): 0.00376

Yes: 4

No: 12

Info\_age(D): 0.013521

Yes: 1

No: 12

Info\_age(D): 0.0052981

Yes: 2

No: 6

Info\_age(D): 0.0067607

Yes: 7

No: 1

Info\_age(D): 0.0045297

Yes: 7

No: 5

Info\_age(D): 0.012248

Yes: 1

No: 8

Info\_age(D): 0.004718

Yes: 1

No: 6

Info\_age(D): 0.0043143

Information Gain=0.19813

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Density için evet/hayir oranlari

Yes: 414

No: 459

Info\_density(D): 0.90763

Yes: 7

No: 9

Info\_density(D): 0.016478

Yes: 18

No: 41

Info\_density(D): 0.054539

Yes: 5

No: 7

Info\_density(D): 0.012248

Information Gain=0.0050414

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Margin için evet/hayir oranlari

Yes: 114

No: 22

Info\_margin(D): 0.090456

Yes: 52

No: 353

Info\_margin(D): 0.23331

Yes: 191

No: 89

Info\_margin(D): 0.26309

Yes: 72

No: 43

Info\_margin(D): 0.11424

Yes: 15

No: 9

Info\_margin(D): 0.023861

Information Gain=0.27099

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Shape için evet/hayir oranlari

Yes: 45

No: 50

Info\_shape(D): 0.098761

Yes: 38

No: 186

Info\_shape(D): 0.15327

Yes: 326

No: 104

Info\_shape(D): 0.35749

Yes: 35

No: 176

Info\_shape(D): 0.14247

Information Gain=0.24395

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Maximum information gain=0.33026

Attribute: BIRADS

Root: BIRADS

6 80 4 5 3 1

6 60 3 5 3 1

6 51 4 4 3 1

6 41 2 1 3 0

6 71 4 4 3 1

6 68 4 3 3 1

6 76 3 1 3 0

6 63 1 1 3 0

6 40 4 3 4 1

6 72 4 3 3 1

6 41 3 3 2 1

ok: 0

ok: 1

Attribute: Margin

ok: 0

6 80 4 5 3 1

6 60 3 5 3 1

6 51 4 4 3 1

6 41 2 1 3 0

6 71 4 4 3 1

6 68 4 3 3 1

6 76 3 1 3 0

6 63 1 1 3 0

6 40 4 3 4 1

6 72 4 3 3 1

6 41 3 3 2 1

count =

11

MATLAB Code for K-Means

%K refers to the number of groups

%We have to know the K! Three clusters = (K=3) Since K=3 I have to identify

%three center points for these clusters

%Sometimes we can pick three points arbitrarily from the dataset as the

%center for each clusters. Or it can be a random number also!

%We will use Rectilinear distance

clc;

clear all;

close all;

import java.util.ArrayList

%Importing the data

data = readmatrix('mammographic\_masses\_new4.csv');

%Split the data as train and test

%K-Fold Cross Validation

%Data has been split up into 10 K-Fold Cross Validation, Severity is the

%class attribute.

cv = cvpartition(data(:, 6),'KFold',10);

idxTest = test(cv, 1); %Test indices for cross-validation

idxTraining = training(cv, 10); %Training indices for cross-validation

%Separate to training and test data

dataTrain = data(~idxTraining,:);

dataTest = data(idxTest,:);

%Splitting the data into attributes

BIRADS = data(:, 1);

Age = data(:, 2);

Shape = data(:, 3);

Margin = data(:, 4);

Density = data(:, 5);

Severity = data(:, 6);

%point eklenir

point = zeros(960,6);

point(:,1) = BIRADS;

point(:,2) = Age;

point(:,3) = Shape;

point(:,4) = Margin;

point(:,5) = Density;

point(:,6) = Severity;

%ClusterArray

ClusterArray=zeros(960, 1);

ClusterOldArray=zeros(960, 1);

%K=3 olsun. point-30 point-100 point-150 benim geçici merkezlerim

center1 = point(30,:);

center2 = point(100,:);

center3 = point(150,:);

clusterRepeat1=0;

clusterRepeat2=0;

clusterRepeat3=0;

clusterSum1=0;

clusterSum2=0;

clusterSum3=0;

for i=1:960

distmean1=point(i,:)-center1;

sumDistMean1=sum(abs(distmean1));

distmean2=point(i,:)-center2;

sumDistMean2=sum(abs(distmean2));

distmean3=point(i,:)-center3;

sumDistMean3=sum(abs(distmean3));

A=min(sumDistMean1, sumDistMean2);

A=min(A, sumDistMean3);

%clusterSum1,2,3 represents how many times we have seen 1, 2, or 3.

if(A == sumDistMean1)

ClusterOldArray(i)=1;

clusterRepeat1=clusterRepeat1+1;

clusterSum1=clusterSum1+point(i,:);

elseif(A == sumDistMean2)

ClusterOldArray(i)=2;

clusterRepeat2=clusterRepeat2+1;

clusterSum2=clusterSum2+point(i,:);

elseif(A == sumDistMean3)

ClusterOldArray(i)=3;

clusterRepeat3=clusterRepeat3+1;

clusterSum3=clusterSum3+point(i,:);

end

end

center1=clusterSum1/clusterRepeat1;

center2=clusterSum2/clusterRepeat2;

center3=clusterSum3/clusterRepeat3;

isChanged=1;

while(isChanged==1)

isChanged=0;

for i=1:960

distmean1=point(i,:)-center1;

sumDistMean1=sum(abs(distmean1));

distmean2=point(i,:)-center2;

sumDistMean2=sum(abs(distmean2));

distmean3=point(i,:)-center3;

sumDistMean3=sum(abs(distmean3));

A=min(sumDistMean1, sumDistMean2);

A=min(A, sumDistMean3);

%clusterSum1,2,3 represents how many times we have seen 1, 2, or 3.

if(A == sumDistMean1)

ClusterArray(i)=1;

if(ClusterArray(i)~=ClusterOldArray(i))

isChanged=1;

ClusterOldArray(i)=ClusterArray(i);

end

clusterRepeat1=clusterRepeat1+1;

clusterSum1=clusterSum1+point(i,:);

elseif(A == sumDistMean2)

ClusterArray(i)=2;

if(ClusterArray(i)~=ClusterOldArray(i))

isChanged=1;

ClusterOldArray(i)=ClusterArray(i);

end

clusterRepeat2=clusterRepeat2+1;

clusterSum2=clusterSum2+point(i,:);

elseif(A == sumDistMean3)

ClusterArray(i)=3;

if(ClusterArray(i)~=ClusterOldArray(i))

isChanged=1;

ClusterOldArray(i)=ClusterArray(i);

end

clusterRepeat3=clusterRepeat3+1;

clusterSum3=clusterSum3+point(i,:);

end

end

center1=clusterSum1/clusterRepeat1;

center2=clusterSum2/clusterRepeat2;

center3=clusterSum3/clusterRepeat3;

end

center1

center2

center3

Çıktı:

center1 =

4.4629 66.1540 3.1452 3.2224 2.9258 0.6398

center2 =

3.9885 30.4962 1.9069 1.4846 2.9154 0.0938

center3 =

4.1360 47.5350 2.4560 2.3220 2.9055 0.3118